Introduction:

In this activity, you will complete a Card class that will be used to create card objects. Think about card games you've played. What kinds of information do these games require a card object to "know"? What kinds of operations do these games require a card object to provide?

Exploration:

Now think about implementing a class to represent a playing card. What instance variables should it have? What methods should it provide? Discuss your ideas for this Card class with classmates.

Read the partial implementation of the Card class. As you read through this class, you will notice the use of the @Override annotation before the toString method. The Java @Override annotation can be used to indicate that a method is intended to override a method in a superclass. In this example, the Object class's toString method is being overridden in the Card class. If the indicated method doesn't override a method, then the Java compiler will give an error message.

Exercises:

- Complete the implementation of the provided Card class. You will be required to complete:
  1. a constructor that takes two String parameters that represent the card's rank and suit, and an int parameter that represents the point value of the card;
  2. accessor methods for the card's rank, suit, and point value;
  3. a method to test equality between two card objects; and
  4. the toString method to create a String that contains the rank, suit, and point value of the card object. The string should be in the following format:

     rank of *suit* (point value = *pointValue*)

- Once you have completed the Card class, add code to Main.java to test everything. Create three Card objects and test each method for each Card object.

  HINT: The matches method should only return true if all three conditions are true. There is more than one way to complete this, but since we haven't yet used if statements, think about what java will return from a comparison:

  ```
  && (and), || (or), == (equals), and str1.equals(str2) all return either true or false
  ```

  This means that the entire check can be placed in the return statement! We will study this further in the next unit.

The remaining activities of the Elevens Lab can be found here. (These may or may not be completed later this year)